User-Representative Feature Selection for Keystroke Dynamics

Eesa Al Solami, Colin Boyd, Andrew Clark and Irfan Ahmed
Information Security Institute, Queensland University of Technology
GPO Box 2434, Brisbane 4001, Queensland, Australia
e.alsolami@student.qut.edu.au
{c.boyd, a.clark, irfan.ahmed}@qut.edu.au

***Abstract*—Continuous user authentication with keystroke dynamics uses characters sequences as features. Since users can type characters in any order, it is imperative to find character sequences (n-graphs) that are representative of user typing behavior. The contemporary feature selection approaches do not guarantee selecting frequently-typed features which may cause less accurate statistical user-representation. Furthermore, the selected features do not inherently reflect user typing behavior. We propose four statistical-based feature selection techniques that mitigate limitations of existing approaches. The first technique selects the most frequently occurring features. The other three consider different user typing behaviors by selecting: n-graphs that are typed quickly; n-graphs that are typed with consistent time; and n-graphs that have large time variance among users. We use Gunetti's keystroke dataset and k-means clustering algorithm for our experiments. The results show that among the proposed techniques, the most-frequent feature selection technique can effectively find user-representative features. We further substantiate our results by comparing the most-frequent feature selection technique with three existing approaches (popular Italian words, common n-graphs, and least frequent n-graphs). We find that it performs better than the existing approaches after selecting a certain number of most-frequent n-graphs.**

***keywords*—feature selection, keystroke dynamics, 2-graphs**

I. Introduction

Continuous user authentication schemes based on keystroke dynamics can be classified as either static or continuous [9]. Static approaches analyze typing behavior of a fixed predefined set of characters (such as a password) for authentication. They are more robust than simple password-matching but they do not detect changes to the initial authorized user later in the session. Continuous authentication approaches, in contrast, monitor and verify the user throughout the computer session.

Continuous authentication approaches with keystroke dynamics use sequences of characters that users type during a session as distinguished features. Since users can type characters in any sequence during a session, continuous authentication approaches require selection of multiple features that are representative of user typing behavior. The n-graph is a popular feature among existing continuous authentication schemes. It is the time interval between the first and the last of n subsequent key-presses. Existing approaches use n-graph (feature) selection techniques to obtain user-representative features that include: popular-word selection (variable length n-graphs that are popular in a language; for instance, *or* or *of*) [7]; common n-graphs selection (often typed by all the users of a system) [3] and the least-frequent n-graphs (least frequently typed by all users of a system) [1].

While these choices exploit the occurrences of n-graphs, their selection criteria do not guarantee features with strong statistical significance. Furthermore, their selected features do not inherently incorporate user typing behavior.

This paper proposes four statistical-based feature selection techniques that overcome some limitations of existing ones. The first is simply the most frequently typed n-graphs selection technique; it selects a certain number of highly occurring n-graphs which we expect to have highest statistical significance. The other three encompasses user's different typing behaviors that include:

1) The quickly-typed n-graph selection technique; it obtains n-graphs that are typed quickly. It computes the average of n-graphs representing their usual typing time and then, selects the n-graphs having least typing time.
2) The time-stability typed n-graph selection technique; it selects the n-graphs that are typed with consistent time. It computes the standard deviation of n-graphs representing the variance from their average typing time and then selects the n-graphs having least variance.
3) The time-variant typed n-graph selection technique; it selects the n-graphs that are typed with noticeably different time. It computes the standard deviation of n-graphs among all users representing the variance from their average typing time and then, selects the n-graphs having large variance.

For evaluating the proposed techniques, we analyze whether selected features are user representative (or reflecting a normal typing pattern of a user). For this, we use 2-graphs (sequence of two characters) as features in this paper because they are the basic element of n subsequent key-presses and occur more frequently than general n-graphs. They are also used by Gunetti et al. [3], Monrose et al. [8], Dowland et al. [2], [1] for their continuous user authentication schemes. Moreover, we use clustering (k-means) algorithm to find out whether the 2-graphs (selected by our techniques) are user representative. The notion is user typing data should be grouped into one clus-

ter using user representative features since clustering naturally group data that share similar behavior (based on features).

The paper is organised as follows. Section II discusses the feature selection techniques in the related work pertaining to the keystroke dynamics. Section III explains the proposed feature selection techniques followed by their evaluation methodology in Section IV and their experimental results in Section V. Section VI concludes the paper.

II. Related work

Dowland et al.[1] collected the typing samples of five users by monitoring their regular computer activities, without any particular constraints being imposed on them such as asking users to type predefined set of words. They selected the features (2-graphs only) that occurred least number of times across the collected typing samples. They use keystroke latency which is the elapsed time between the release of the first key and the press of the second key. They build user profiles by computing the mean and standard deviation of 2-graphs latency. They achieved correct acceptance rates in the range of 60%.

Unlike Dowland et al., Gunetti et al.[3] avoided using the 2-graphs and 3-graphs latencies directly as features. Instead, they used latencies that determine the relative ordering of different 2-graphs/3-graphs. They extracted the 2-graphs and 3-graphs that are common between two samples and found the difference between them. For this, they devised a distance metric to measure the distance between the two-orderings of 2-graphs and 3-graphs between two samples. In order to identify the user of an unknown sample, they compare it with all the samples of the users by computing the distance between them. The user's sample with least distance is deemed to be the user of the unknown sample. They reported 95% accuracy.

Rajkumar and Sim [7] selected popular English words such as the, or, to, you as features. They showed that many fixed strings qualify as good candidates and identified the user as soon as he typed any of the fixed strings. They proved that these words can be used to discriminate users effectively.

III. Proposed feature selection techniques

We proposed four statistical-based feature selection techniques. Their details are as below.

A. Most frequently typed n-graph selection

The dataset that contains n-graphs with higher frequencies should receive a higher score or rate in terms of frequency and 2-graphs is basically the most frequent of n-graphs. The weight for 2-graphs is assigned to each 2-graphs in the dataset, which depends on the number of occurrences or frequencies of the 2-graphs in the dataset. The simplest approach is to assign the weight to be equal to the number of occurrences of the 2-graphs in dataset d. This weighting scheme is referred to as the 2-graphs frequency. Each 2-graph may occur in d more than once in each sample s, and then average or standard deviation will be calculated and tested independently in order to see the impact of these statistics on the process of distinguishing the users.

B. Quickly-typed n-graph selection

The aim of this technique is to identify and extract the 2-graphs that are typed by users very quickly. The AVG function was used in this technique to represent the behavior of the user's typing of 2-graphs; however, the user has actually typed the same 2-graphs more than once in the one sample. The average of the 2-graphs was calculated, based on the average for all the user samples AVG 2-graphs $\{U_i\,,\,S_j\}$ where U_i indicates the user number and S_j indicates the sample number to determine the 2-graphs with the smallest average. Therefore, the 2-graph list was reordered based on the lowest AVG time. The top of the 2-graph list means that the user typed these 2-graphs very quickly.

C. Time-stability typed n-graph selection

The aim of this technique is to identify and extract the 2-graphs typed by U_i consistently most of the time. STD was used here to test the typing stability of 2-graphs for every $\{U_i\,,\,S_j\}$ that the STD 2-graphs$\{U_i\,,\,S_j\}$. Then the AVG for 2-graphs was calculated and the 2-graphs list was reordered based on the smallest AVG. Being placed at the top of the 2-graphs list means that these 2-graphs have more stable user typing than the other 2-graphs.

D. Time-variant typed n-graph selection

The aim of this technique is to identify and extract the 2-graphs that are typed by users very differently. First, for each 2-graphs in the set d, the AVG time of the 2-graphs has been calculated per u that shows by AVG 2-graphs $\{U_i\}$ and then the STD was calculated for each 2-graphs among U_i in order to see the typing variances between users for that 2-graphs. The 2-graphs list was then reordered, based on the largest STD, in order to identify the top 2-graphs that the user typed differently. Thus, if the STD of the 2-graphs is very high, this means that the users typing the 2-graphs are similar; however, if the STD of the 2-graphs is very high, then the users type the 2-graphs differently and can be easily distinguished. Therefore, the 2-graphs list was reordered based on the highest STD time. The top of the 2-graphs list means that the user typed these 2-graphs very differently.

IV. Evaluation Methodology

Figure 1 shows two main phases: the feature selection phase and the evaluation phase and the descriptions of the two phases are described below in detail.

A. Selecting candidate features

This phase has three sub-phases including: pre-processing dataset, apply the feature selection techniques and then, extract the dataset after preprocessed and with the candidate features. The three sub-phases are described next.

1) Pre-processing: The pre-processing sub-phase is defined by transforming the numbers in the keystroke raw data into Italian characters as contained by the Gunetti's dataset with the duration times. The keystroke raw data contain data as a decimal ASCII and we were

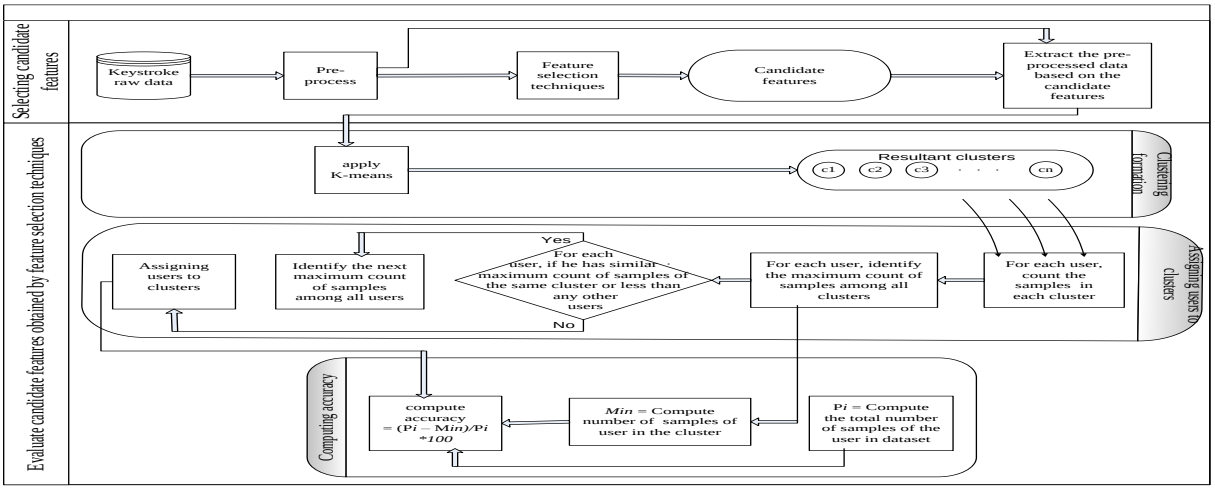

Figure 1. Evaluation methodology for feature selection techniques

transformed to the real characters in Italian, based on the ISO/IEC 8859-16 standard Italian character set. However, a strange character was found for some users and this strange character is "zero" in the ASCII code. Therefore, when trying to transform it into an Italian character, it came up as "null value". Thus, the character is not understandable, and if removed from the data sample that included it, it affected the sequence of the time for the rest of the characters. Therefore, it was decided to remove the data of these users. After that, the feature selection techniques phase took place, which is the main aim of this paper.

2) Feature selection techniques: This sub-phase is selecting and extracting some features that could represent the user's typing pattern based on different techniques. These techniques can be selected based on different typing user's behavior such as when the user typed quickly. Also, these techniques can be selected based on statistical measures such on the less frequent n-graph or most frequent n-graph. Then, the output of the feature selection techniques will be some candidate features that could represent the user's typing pattern.
3) Extract the preprocessed data based on the candidate features: This sub-phase will go to the preprocessed data and extract the relevant dataset that including the users and samples based on the candidate features that have been identified from the previous sub-phase. Then, the relevant dataset will be used and tested in the next following evaluation phase.

B. Evaluate candidate features (obtained by feature selection techniques)

This phase has three sub-phases and the descriptions of them are described below in detail.

1) Cluster formation: Clustering in general is naturally grouped samples that share similar behaviors which can be useful to partition the user samples into subsets whose in-class members are "similar" in the identified features and whose cross-class members are dissimilar as in the corresponding sense. Clustering algorithms have been successfully used for evaluating and testing different features of keystroke dynamics [5][6]. k-means is the most well-known clustering algorithm and unsupervised learning algorithms that solve the known clustering problem. Thus, the k-means was used to evaluate the feature selection techniques. It applied in the dataset in order to divide the user's samples into different clusters.

The procedure is to group a given user's samples through a certain number of clusters (assumed k clusters) fixed a prior based on the total number of users. The main idea is to define k-centroids, one for each cluster. These centroids should be placed in a clever way, because a different location causes a different result. Given a set of observations(S_i), where each observation in our case is a user sample, k-means clustering C_n aims to partition the (S_i) observations into $c_n$sets there is fourmula here so as to minimize the within-cluster sum of squares. Some distance types can be used with clustering, but here in this paper, the evaluation is based on the two most popular distances, including Euclidean distance and city-block distance.

2) Assigning Users: Due to the ambiguity of the clustering result in that some clusters contain samples from different users, it is necessary to rely on criteria that is able to extract relevant information about distinguishing users. The output of the k-means technique needs to design criteria that is able to assign each user to a different cluster. The decision of assigning each user to different cluster is based on the frequency of the samples to the cluster. For example, if cluster 1 has been assigned by user 1 with 5 samples and by user 2 with 10 samples, then the decision will be assign user 2 to cluster 1. However, if the frequency of the samples are similar for two users and assigned to one cluster, then the cluster will

not assigned to any one of those users. They will assigned based on the second most frequency samples of them.

3) Computing Accuracy: Accuracy rate was used in this classification system in order to measure and evaluate the measurement of the system

$$Accuracy = \left(\frac{P_i - M_{in}}{P_i}\right) \times 100$$

Where:

$$P_i = \sharp \text{ of samples of } U_i \text{ in the dataset}$$

$$M_{in} = \sharp \text{ of samples of } U_i \text{ in } C_n$$

V. Experiments

This section divided to 4 sub-sections:

A. Dataset

We obtained the dataset from Gunetti et al.[3]. They gathered it over a 6-month period. It contains 15 samples from each of the 40 different users. This dataset is popular in the area of keystroke dynamics and recognised by some researchers and also, has been used for evaluation by some of them [4]. There was no constraint on the users as they are free to type whatever they like. The users were asked to provide no more than one sample of 700-900 characters per day, but could provide it at any time of the day. They were allowed to type whatever they liked, except that they were not to type the same text for more than one sample. The sample was collected using a web-based form that recorded the ASCII characters and associated key-press times. Not all users gave permission for their samples to be released to a third party; therefore, the provided dataset contained data for only 21 users and each user has 15 typing samples. However, as we explained in section IV-A that some user's data have been removed and only 14 users with 15 samples used in this study.

B. Experimental settings

For the sake of conciseness, only 60 (2-graphs) that were the most frequent in the dataset were considered, since 100% accuracy was reached with these numbers of features. Then, the proposed feature selection techniques were applied on the 60 2-graphs list. Therefore, each feature selection technique will produce the same list of 60 (2-graphs), but in a new order. For sake of conciseness, the new list of 60 (2-graphs) were divided into 6 groups in order to see the impact of the evaluation of each group cumulatively by adding one group by one. Each group contained 10 (2-graphs), for example, group 10 represented the first 10 (2-graphs) from the new list that was produced after applying the feature selection technique, and group 60 represented the last 10^{th} (2-graphs) from the new list.

C. Experimental results

The result has been presented the accuracy of the selected features based on cumulatively adding one group of features one by one. The purpose of presenting the accuracy result of adding the number of selected features cumulatively is to see the impact of the proposed feature selection with different number of features. For all the figures, the numbers in the horizontal line represent the number of selected features of 2-graphs. Percentages in the vertical line represent the system of accuracy percentage of the user samples that were correctly classified to the right user.

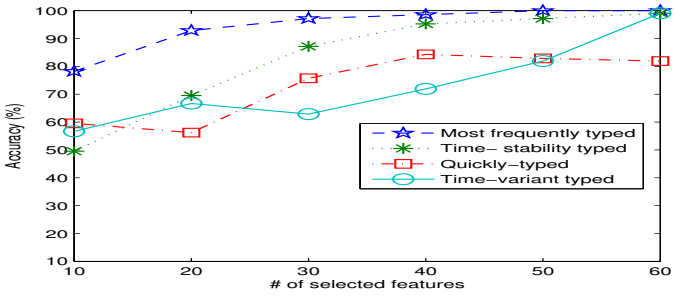

Figure 2. Comparison between proposed feature selection techniques based on # of selected features cumulatively

Figure 2 shows the comparison between all of the proposed feature selection techniques where adding the number of selected features cumulatively. The comparative analysis of all of the proposed feature selection techniques of 2-graphs demonstrates that the most frequently typed n-graph selection technique has the highest accuracy percentage which lead to show the user representative effectively because of their highest statistical significance. Furthermore, modeling the user behavior by the time-stability typed n-graph selection technique is still promising to represent the user typing effectively. This is due to the consistent time of the selected n-graph which represent the normal user's typing behavior. However, for a less number of 2-graphs, the most frequent 2-graphs technique obtained a much better accuracy percentage compared to other proposed feature selection techniques.

Also, this study compard between city block distance and city block distance. The result shows that the city block distance is slightly better than Euclidean distance. Also, in case of using city block distance, we compared between different statistics measures including $AVG, STD, AVG+STD$ when the durations of 2-graphs occurred more than once in the user's sample. The result shows in figure 3 that the accuracy result of based on AVG is quite similar to $AVG + STD$, after selecting 20 features of 2-graphs cumulatively. However, the 2-graphs that were calculated based on $AVG+STD$ required more computation than the 2-graphs that were only calculated based on AVG.

D. Comparison with existing feature selection techniques

Regarding the feature of the most frequent English words, the most frequent Italian words are analysed and tested as this dataset is based on Italian words such as non, di, che, la, and il. Table 1 shows the comparison of classification accuracy between the feature of the five most frequent Italian words with the five most frequent 2-graphs. The result of the classification accuracy is quite similar if the analysis is done individually per word or 2-graph. However, if the analysis is

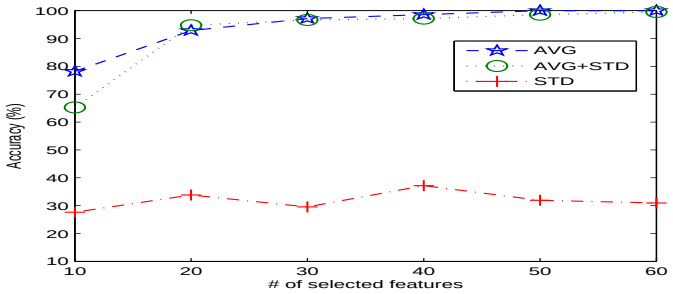

Figure 3. Comparison between different statistics for the most frequent 2-graphs technique based on # of selected features cumulatively

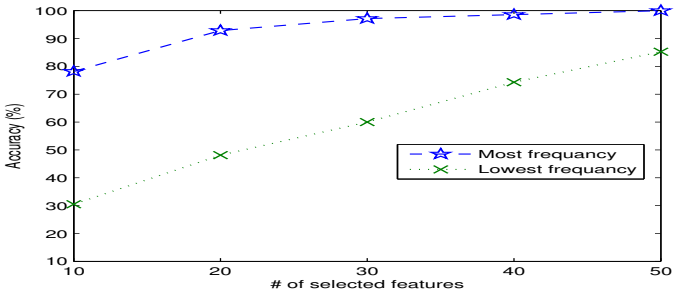

Figure 4. Comparison between the most and least frequent 2-graphs based on # of selected features cumulatively

Table I
COMPARISON OF ITALIAN WORDS AND MOST FREQUENT 2-GRAPHS

Italian words	Classifier accuracy (in percentages) (B)	Most frequent 2-graphs	Classifier accuracy (in percentages) (A)	Improvement A-B
non	40.00	co	34.28	-5.72
di	29.52	to	33.80	4.28
che	28.09	re	29.04	0.95
la	28.57	er	28.57	0
il	23.33	on	23.33	0
combining all of them together	60.48	combining all of them together	64.76	4.28

done by combining all of the five most frequent Italian words and the five most frequent 2-graphs, it can be seen that the five most frequent 2-graphs are better than the five most frequent 2-graphs in terms of classification accuracy.

Table 2 shows the comparison of classification accuracy between the feature of the common 2 and 3 graphs and the most frequent 2-graphs. In this dataset, the count of the common 2 and 3 graphs was 19, and to get a comparable result, the most 19 frequent 2-graphs were compared with the 19 common 2 and 3 graphs. The result of the common 2 and 3 graphs is slightly better in terms of classification accuracy. If the number of the most frequent 2-graphs is increased to 20, a similar result is obtained for the common 2 and 3 graphs feature. Moreover, the common 2 and 3 graphs feature do not reach 100% classification accuracy in this dataset and this percentage cannot be obtained based on the most frequent 2-graphs feature in taking 50 values of the feature in account.

Table II
COMPARISON BETWEEN COMMON 2-GRAPHS AND MOST FREQUENT 2-GRAPHS

Most frequent 2-graphs/common n-graphs	Number of common 2- and 3-graphs	Number of most frequent 2- graphs				
	19	19	20	30	40	50
Classifier accuracy (in percentages)	95.71	90.95	95.23	97.14	98.57	100

Furthermore, the feature of the common 2 and 3 graphs is very dependent on all users typing.

Figure 4 exhibits the result of the classification accuracy for both the most frequency 2-graphs and lowest frequency 2-graphs where adding the groups of 2-graphs cumulatively. It shows from the graph that the selecting less than 20 numbers of most frequent 2-graphs performs better than selecting 50 numbers of lowest frequency 2-graphs.

VI. CONCLUSIONS

We proposed four statistical-based feature selection techniques for keystroke dynamics. We use 2-graph (as features) in our experiments and found that the most-frequent 2-graphs technique can represent user's typing patterns effectively because of their highest statistical significance.

Among the other three proposed techniques, n-graph that are typed with consistent time showed promising results. It achieved significantly higher accuracy and even after selecting a certain number of features, it matches the accuracy near to the frequently typed n-graphs.

REFERENCES

[1] P. Dowland, S. Furnell, and M. Papadaki. Keystroke analysis as a method of advanced user authentication and response. *Security in the Information Society: Visions and Perspectives*, page 215, 2002.

[2] P. Dowland, H. Singh, and S. Furnell. A preliminary investigation of user authentication using continuous keystroke analysis. In *Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas*, pages 27–28, 2001.

[3] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, 8(3):312–347, 2005.

[4] K. Hempstalk. *Continuous Typist Verification using Machine Learning*. PhD thesis, The University of Waikato, 2009.

[5] S. Hocquet, J. Ramel, and H. Cardot. User Classification for Keystroke Dynamics Authentication. *Lecture Notes in Computer Science*, 4642:531, 2007.

[6] J. Hu, D. Gingrich, and A. Sentosa. A k-Nearest neighbor approach for user authentication through biometric keystroke dynamics. In *Proceedings of the IEEE International Conference on Communications*, pages 1556–1560, 2008.

[7] R. Janakiraman and T. Sim. Keystroke Dynamics in a General Setting. *Lecture Notes in Computer Science*, 4642:584, 2007.

[8] F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 48–56. ACM New York, NY, USA, 1997.

[9] F. Monrose and A.D. Rubin. Keystroke dynamics as a biometric for authentication. *FUTURE GENER COMPUT SYST*, 16(4):351–359, 2000.